Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	18
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 17x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f18346-i-p

TABLE 4-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18326	PIC16(L)F18346	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
Bank 28													
CPU CORE REGISTERS; see Table 4-2 for specifics													
E0Ch	—	—		Unimplemented								—	—
E0Dh	—	—		Unimplemented								—	—
E0Eh	—	—		Unimplemented								—	—
E0Fh	PPSLOCK			—	—	—	—	—	—	—	PPSLOCKED	---- ---0	---- ---0
E10h	INTPPS			—	—	—	INTPPS<4:0>					---0 0010	---u uuuu
E11h	T0CKIPPS			—	—	—	T0CKIPPS<4:0>					---0 0010	---u uuuu
E12h	T1CKIPPS			—	—	—	T1CKIPPS<4:0>					---0 0101	---u uuuu
E13h	T1GPPS			—	—	—	T1GPPS<4:0>					---0 0100	---u uuuu
E14h	CCP1PPS			—	—	—	CCP1PPS<4:0>					---1 0011	---u uuuu
E15h	CCP2PPS			—	—	—	CCP2PPS<4:0>					---1 0101	---u uuuu
E16h	CCP3PPS			—	—	—	CCP3PPS<4:0>					---0 0010	---u uuuu
E17h	CCP4PPS	X	—	—	—	—	CCP4PPS<4:0>					---1 0001	---u uuuu
		—	X	—	—	—	CCP4PPS<4:0>					---0 0100	---u uuuu
E18h	CWG1PPS			—	—	—	CWG1PPS<4:0>					---0 0010	---u uuuu
E19h	CWG2PPS			—	—	—	CWG2PPS<4:0>					---0 0010	---u uuuu
E1Ah	MDCIN1PPS			—	—	—	MDCIN1PPS<4:0>					---1 0010	---u uuuu
E1Bh	MDCIN2PPS			—	—	—	MDCIN2PPS<4:0>					---1 0101	---u uuuu
E1Ch	MDMINPPS			—	—	—	MDMINPPS<4:0>					---1 0011	---u uuuu
E1Dh	SSP2CLKPPS	X	—	—	—	—	SSP2CLKPPS<4:0>					---1 0100	---u uuuu
		—	X	—	—	—	SSP2CLKPPS<4:0>					---0 1111	---u uuuu
E1Eh	SSP2DATPPS	X	—	—	—	—	SSP2DATPPS<4:0>					---1 0101	---u uuuu
		—	X	—	—	—	SSP2DATPPS<4:0>					---0 1101	---u uuuu
E1Fh	SSP2SSPPS	X	—	—	—	—	SSP2SSPPS<4:0>					---0 0000	---u uuuu
		—	X	—	—	—	SSP2SSPPS<4:0>					---0 0001	---u uuuu
E20h	SSP1CLKPPS	X	—	—	—	—	SSP1CLKPPS<4:0>					---1 0000	---u uuuu
		—	X	—	—	—	SSP1CLKPPS<4:0>					---0 1110	---u uuuu

Legend: x = unknown, u = unchanged, q = depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.

Note 1: Only on PIC16F18326/18346.

2: Register accessible from both User and ICD Debugger.

6.10 Start-up Sequence

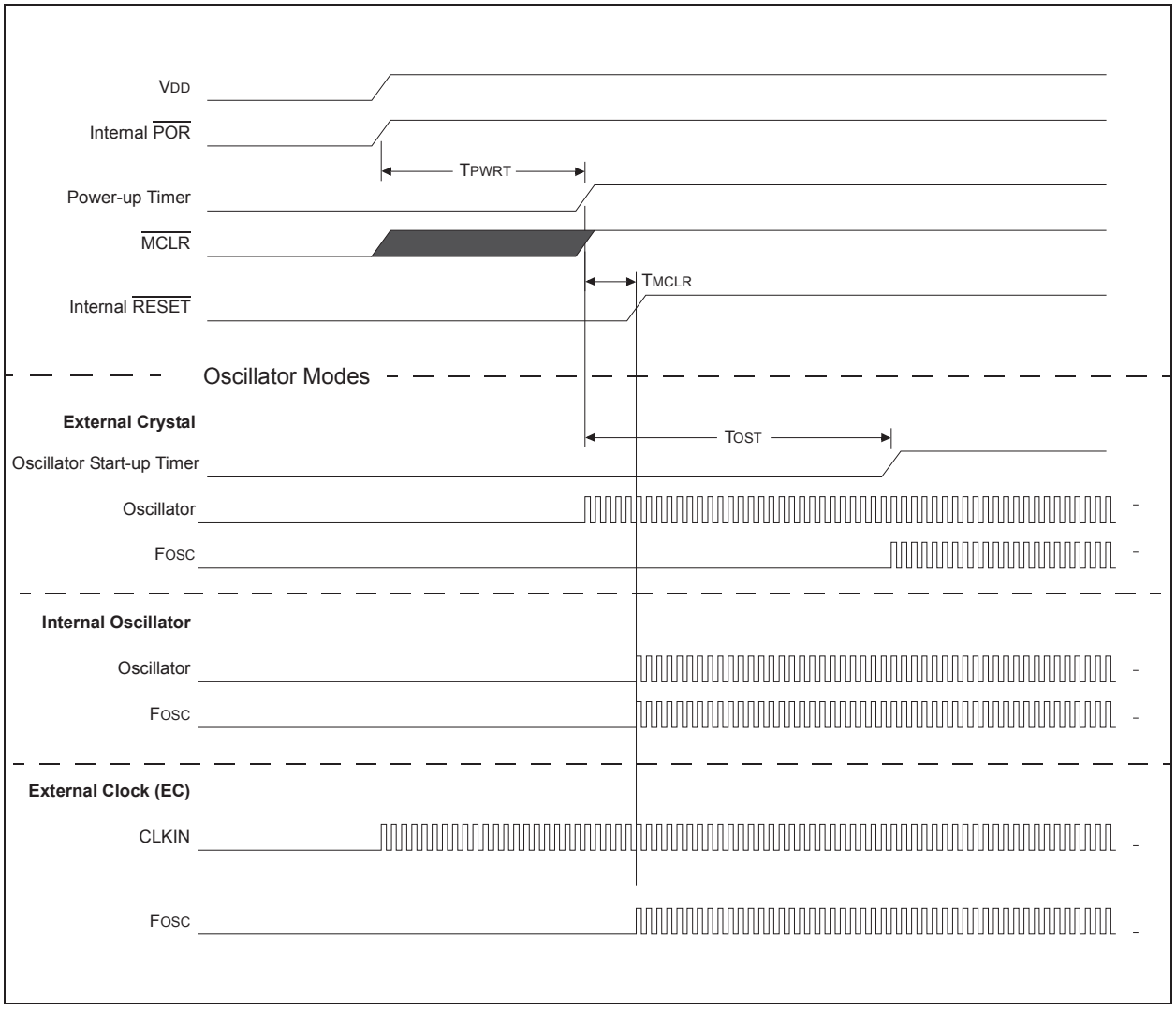
Upon the release of a POR or BOR, the following must occur before the device will begin executing:

1. Power-up Timer runs to completion (if enabled).
2. $\overline{\text{MCLR}}$ must be released (if enabled).
3. Oscillator start-up timer runs to completion (if required for oscillator source).

The total time out will vary based on oscillator configuration and Power-up Timer Configuration. See Section 7.0, Oscillator Module for more information.

The Power-up Timer and oscillator start-up timer run independently of $\overline{\text{MCLR}}$ Reset. If $\overline{\text{MCLR}}$ is kept low long enough, the Power-up Timer will expire. Upon bringing $\overline{\text{MCLR}}$ high, the device will begin execution after ten FOSC cycles (see Figure 6-3). This is useful for testing purposes or to synchronize more than one device operating in parallel.

FIGURE 6-3: RESET START-UP SEQUENCE



6.13 Register Definitions: Power Control

REGISTER 6-2: PCON0: POWER CONTROL REGISTER 0

R/W/HS-0/q	R/W/HS-0/q	U-0	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-q/u	R/W/HC-q/u
STKOVF	STKUNF	—	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

Legend:		
HC = Bit is cleared by hardware		HS = Bit is set by hardware
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-m/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	q = Value depends on condition

bit 7 **STKOVF:** Stack Overflow Flag bit
1 = A Stack Overflow occurred
0 = A Stack Overflow has not occurred or has been cleared by firmware

bit 6 **STKUNF:** Stack Underflow Flag bit
1 = A Stack Underflow occurred
0 = A Stack Underflow has not occurred or has been cleared by firmware

bit 5 **Unimplemented:** Read as '0'

bit 4 **$\overline{\text{RWDT}}$:** Watchdog Timer Reset Flag bit
1 = A Watchdog Timer Reset has not occurred or set to '1' by firmware
0 = A Watchdog Timer Reset has occurred (cleared by hardware)

bit 3 **$\overline{\text{RMCLR}}$:** $\overline{\text{MCLR}}$ Reset Flag bit
1 = A $\overline{\text{MCLR}}$ Reset has not occurred or set to '1' by firmware
0 = A $\overline{\text{MCLR}}$ Reset has occurred (cleared by hardware)

bit 2 **$\overline{\text{RI}}$:** RESET Instruction Flag bit
1 = A RESET instruction has not been executed or set to '1' by firmware
0 = A RESET instruction has been executed (cleared by hardware)

bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{\text{BOR}}$:** Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset occurs)

TABLE 6-5: SUMMARY OF REGISTERS ASSOCIATED WITH RESETS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BORCON	SBOREN	—	—	—	—	—	—	BORRDY	76
PCON0	STKOVF	STKUNF	—	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	77
STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	30
WDTCON	—	—	WDTPS<4:0>					SWDTEN	121

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by Resets.

7.2.2.3 Internal Oscillator Frequency Adjustment

The HFINTOSC and LFINTOSC internal oscillators are both factory-calibrated. TH HFINTOSC oscillator can be adjusted in software by writing to the OSCTUNE register (Register 7-3). OSCTUNE does not affect the LFINTOSC frequency.

The default value of the OSCTUNE register is 00h. The value is a 6-bit two's complement number. A value of 1Fh will provide an adjustment to the maximum frequency. A value of 20h will provide an adjustment to the minimum frequency.

When the OSCTUNE register is modified, the HFINTOSC oscillator frequency will begin shifting to the new frequency. Code execution continues during this shift. There is no indication that the shift has occurred.

7.2.2.4 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is a factory-calibrated 31 kHz internal clock source.

The LFINTOSC is the clock source for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM). The LFINTOSC can also be used as the system clock, or as a clock or input source to certain peripherals.

The LFINTOSC is selected as the clock source through one of the following methods:

- Programming the RSTOSC<2:0> bits of Configuration Word 1 to enable LFINTOSC.
- Write to the NOSC<2:0> bits of the OSCCON1 register.

7.2.2.5 Oscillator Status and Manual Enable

The 'ready' status of each oscillator is displayed in the OSCSTAT1 register (Register 7-4). The oscillators can also be manually enabled through the OSCEN register (Register 7-5). Manual enables make it possible to verify the operation of the EXTOSC or SOSC crystal oscillators. This can be achieved by enabling the selected oscillator, then watching the corresponding 'ready' state of the oscillator in the OSCSTAT1 register.

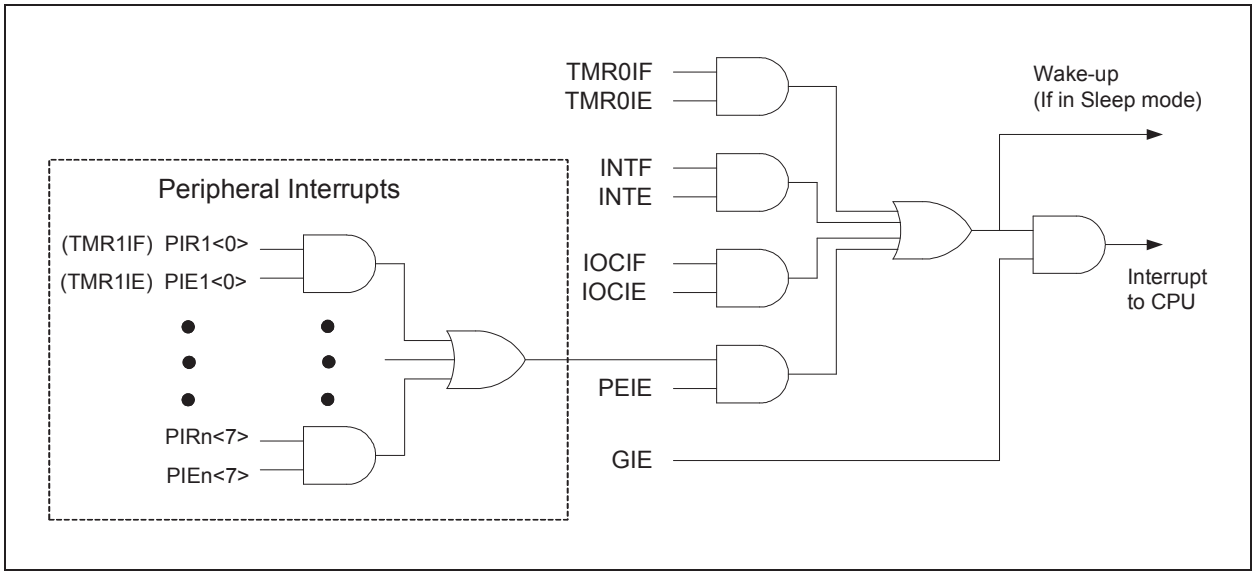
8.0 INTERRUPTS

The interrupt feature allows certain events to preempt normal program flow. Firmware is used to determine the source of the interrupt and act accordingly. Some interrupts can be configured to wake the MCU from Sleep mode.

Many peripherals produce interrupts. Refer to the corresponding chapters for details.

A block diagram of the interrupt logic is shown in Figure 8-1.

FIGURE 8-1: INTERRUPT LOGIC



8.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- GIE bit of the INTCON register
- Interrupt Enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the Interrupt Enable bit of the interrupt event is contained in the PIEx registers).

The PIR1, PIR2, PIR3 and PIR4 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual interrupt enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- Critical registers are automatically saved to the shadow registers (See **"Section 8.5 "Automatic Context Saving"**)
- PC is loaded with the interrupt vector 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack, restoring the saved context from the shadow registers and setting the GIE bit.

For additional information on a specific interrupt's operation, refer to its peripheral chapter.

Note 1:	Individual interrupt flag bits are set, regardless of the state of any other enable bits.
2:	All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.

8.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three or four instruction cycles. The interrupt is sampled during Q1 of the instruction cycle. The actual interrupt latency then depends on the instruction that is executing at the time the interrupt is detected. See Figure 8-2 and Figure 8-3 for more details.

REGISTER 8-3: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **TMR1GIE:** Timer1 Gate Interrupt Enable bit
1 = Enables the Timer1 gate acquisition interrupt
0 = Disables the Timer1 gate acquisition interrupt

bit 6 **ADIE:** Analog-to-Digital Converter (ADC) Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

bit 5 **RCIE:** EUSART Receive Interrupt Enable bit
1 = Enables the EUSART receive interrupt
0 = Disables the EUSART receive interrupt

bit 4 **TXIE:** EUSART Transmit Interrupt Enable bit
1 = Enables the EUSART transmit interrupt
0 = Disables the EUSART transmit interrupt

bit 3 **SSP1IE:** Synchronous Serial Port (MSSP) Interrupt Enable bit
1 = Enables the MSSP interrupt
0 = Disables the MSSP interrupt

bit 2 **BCL1IE:** MSSP1 Bus Collision Interrupt Enable bit
1 = MSSP bus collision interrupt enabled
0 = MSSP bus collision interrupt not enabled

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the Timer2 to PR2 match interrupt
0 = Disables the Timer2 to PR2 match interrupt

bit 0 **TMR1IE:** Timer1 Overflow Interrupt Enable bit
1 = Enables the Timer1 overflow interrupt
0 = Disables the Timer1 overflow interrupt

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

REGISTER 12-13: WPUB: WEAK PULL-UP PORTB REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-4 **WPUB<7:4>**: Weak Pull-up Register bits
1 = Weak Pull-up enabled
0 = Weak Pull-up disabled

bit 3-0 **Unimplemented:** Read as '0'

REGISTER 12-14: ODCONB: PORTB OPEN-DRAIN CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
ODCB7	ODCB6	ODCB5	ODCB4	—	—	—	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-4 **ODCB<7:4>:** PORTB Open-Drain Configuration bits
For RB<7:4> pins, respectively:
1 = Port pin operates as open-drain drive (sink current only)
0 = Port pin operates as standard push-pull drive (source and sink current)

bit 3-0 **Unimplemented:** Read as '0'

REGISTER 15-5: IOCBN: INTERRUPT-ON-CHANGE PORTB NEGATIVE EDGE REGISTER(1)

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-4 **IOCBN<7:4>:** Interrupt-on-Change PORTB Negative Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a negative going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin

bit 3-0 **Unimplemented:** Read as '0'

Note 1: PIC16(L)F18346 only.

REGISTER 15-6: IOCBF: INTERRUPT-ON-CHANGE PORTB FLAG REGISTER(1)

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	U-0	U-0	U-0	U-0
IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HS - Bit is set in hardware

bit 7-4 **IOCBF<7:4>:** Interrupt-on-Change PORTB Flag bits
1 = An enabled change was detected on the associated pin
Set when IOCBPx = 1 and a rising edge was detected on RBx, or when IOCBNx = 1 and a falling edge was detected on RBx.
0 = No change was detected, or the user cleared the detected change.

bit 3-0 **Unimplemented:** Read as '0'

Note 1: PIC16(L)F18346 only.

18.9 Analog Input Connection Considerations

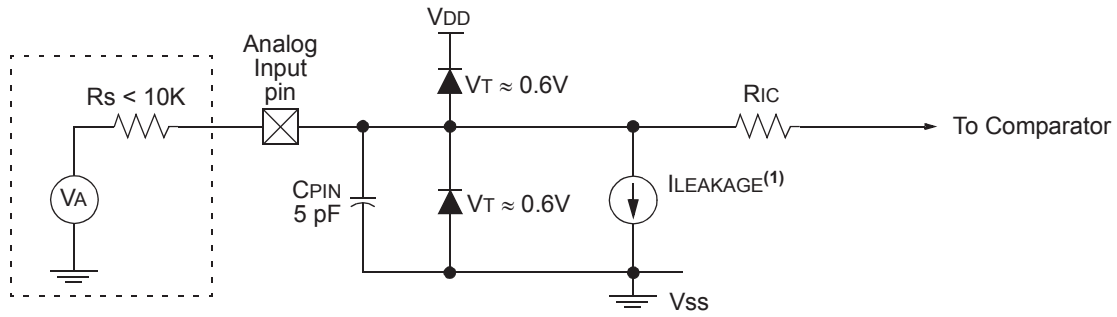
A simplified circuit for an analog input is shown in Figure 18-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, may have very little leakage current to minimize inaccuracies introduced.

Note 1: When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will provide an input based on their level as either a TTL or ST input buffer.

2: Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

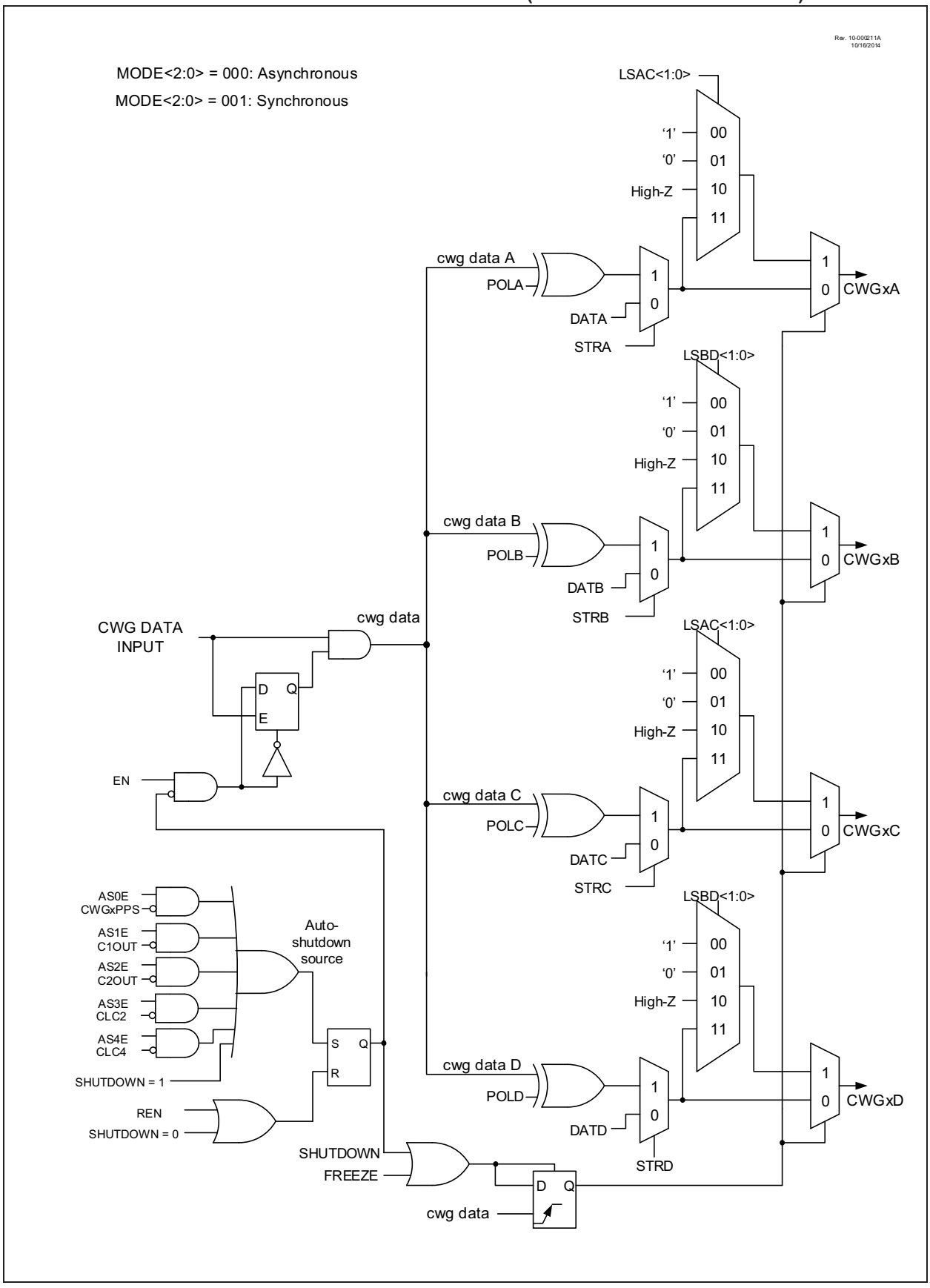
FIGURE 18-3: ANALOG INPUT MODEL



Legend: CPIN = Input Capacitance
ILEAKAGE = Leakage Current at the pin due to various junctions
RIC = Interconnect Resistance
RS = Source Impedance
VA = Analog Voltage
VT = Threshold Voltage

Note 1: See I/O Ports in Table 35-4.

FIGURE 20-10: SIMPLIFIED CWG BLOCK DIAGRAM (OUTPUT STEERING MODES)



20.11 Register Definitions: CWG Control

REGISTER 20-1: CWGxCON0: CWGx CONTROL REGISTER 0

R/W-0/0	R/W/HC-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
EN	LD(1)	—	—	—	MODE<2:0>		
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HS/HC = Bit is set/cleared by hardware

bit 7 **EN:** CWGx Enable bit
- 1 = CWGx is enabled
- 0 = CWGx is disabled

bit 6 **LD:** CWG Load Buffers bit(1)
- 1 = Dead-band count buffers to be loaded on CWG data rising edge following first falling edge after this bit is set.
- 0 = Buffers remain unchanged

bit 5-3 **Unimplemented**: Read as '0'

bit 2-0 **MODE<2:0>**: CWGx Mode bits
- 111 = Reserved
- 110 = Reserved
- 101 = CWG outputs operate in Push-Pull mode
- 100 = CWG outputs operate in Half-Bridge mode
- 011 = CWG outputs operate in Reverse Full-Bridge mode
- 010 = CWG outputs operate in Forward Full-Bridge mode
- 001 = CWG outputs operate in Synchronous Steering mode
- 000 = CWG outputs operate in Asynchronous Steering mode

Note 1: This bit can only be set after EN = 1; it cannot be set in the same cycle when EN is set.

TABLE 22-3: SUMMARY OF REGISTERS ASSOCIATED WITH ADC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	100
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	102
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	107
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	143
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	149
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	155
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	144
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	150
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	157
ADCON0	CHS<5:0>						GO/$\overline{\text{DONE}}$	ADON	244
ADCON1	ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>		245
ADACT	—	—	—	ADACT<4:0>					246
ADRESH	ADRESH<7:0>								247
ADRESL	ADRESL<7:0>								247
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		180
DAC1CON1	—	—	—	DAC1R<4:0>					264
OSCSTAT1	EXTOR	HFOR	—	LFOR	SOR	ADOR	—	PLLR	91

Legend: – = unimplemented read as '0'. Shaded cells are not used for the ADC module.

Note 1: PIC16(L)F18346 only.

2: Unimplemented, read as '1'.

FIGURE 25-2: ON OFF KEYING (OOK) SYNCHRONIZATION

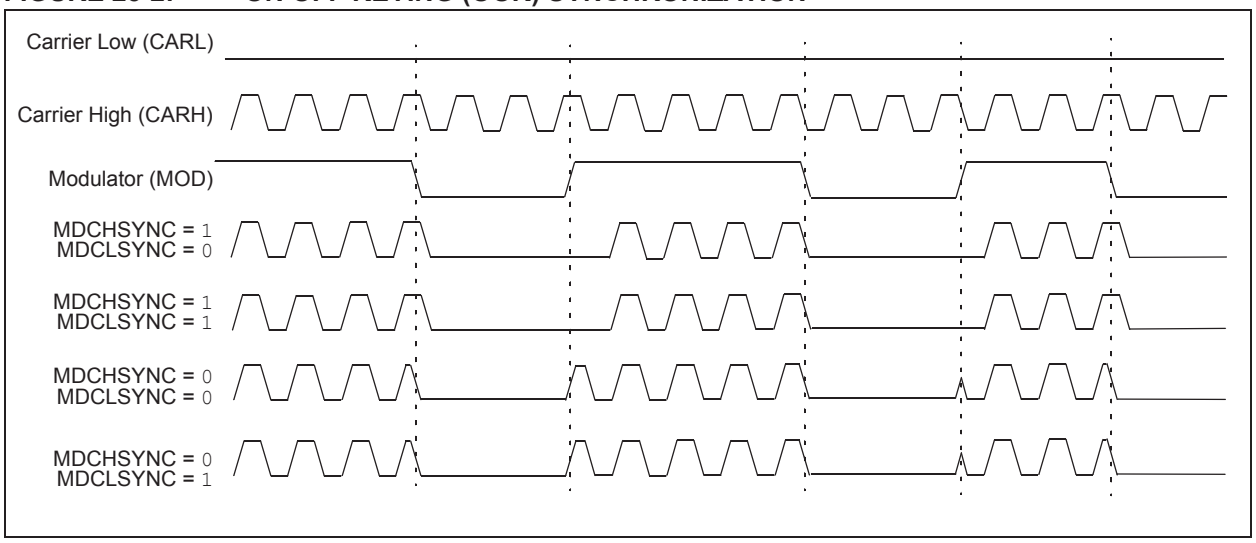


FIGURE 25-3: NO SYNCHRONIZATION (MDSHSYNC = 0, MDCLSYNC = 0)

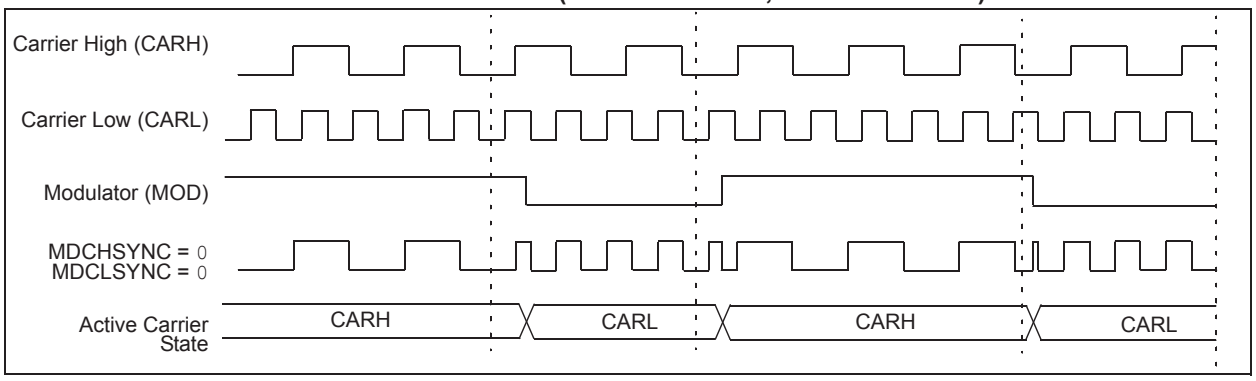
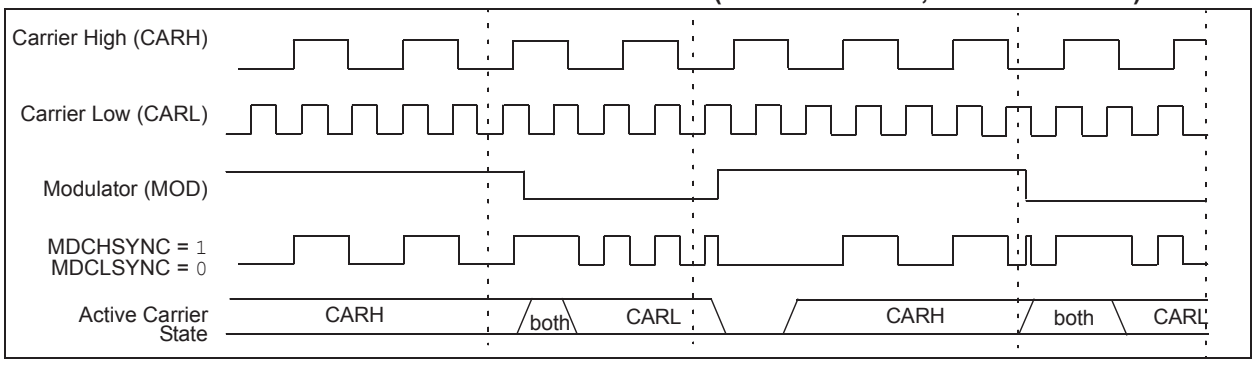


FIGURE 25-4: CARRIER HIGH SYNCHRONIZATION (MDSHSYNC = 1, MDCLSYNC = 0)



29.2.2 TIMER1/3/5 MODE RESOURCE

Timer1/3/5 must be running in Timer mode or Synchronized Counter mode for the CCP module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

See **Section 27.0 "Timer1/3/5 Module with Gate Control"** for more information on configuring Timer1/3/5.

Note:	Clocking Timer1/3/5 from the system clock (FOSC) should not be used in Capture mode. In order for Capture mode to recognize the trigger event on the CCPx pin, Timer1/3/5 must be clocked from the instruction clock (FOSC/4) or from an external clock source.

29.2.3 SOFTWARE INTERRUPT MODE

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE interrupt enable bit of the PIE4 register clear to avoid false interrupts. Additionally, the user should clear the CCPxIF interrupt flag bit of the PIR4 register following any change in Operating mode.

29.2.4 CCP PRESCALER

There are four prescaler settings specified by the CCPxMODE<3:0> bits of the CCPxCON register. Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. Any Reset will clear the prescaler counter.

Switching from one capture prescaler to another does not clear the prescaler and may generate a false interrupt. To avoid this unexpected operation, turn the module off by clearing the CCPxCON register before changing the prescaler. Example 29-1 demonstrates the code to perform this function.

EXAMPLE 29-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
BANKSEL CCPxCON     ;Set Bank bits to point
                    ;to CCPxCON
CLRF    CCPxCON     ;Turn CCP module off
MOVLW   NEW_CAPT_PS ;Load the W reg with
                    ;the new prescaler
                    ;move value and CCP ON
MOVWF   CCPxCON     ;Load CCPxCON with this
                    ;value
```

29.2.5 CAPTURE DURING SLEEP

Capture mode depends upon the Timer1/3/5 module for proper operation. There are two options for driving the Timer1/3/5 module in Capture mode. It can be driven by the instruction clock (FOSC/4), or by an external clock source.

When Timer1/3/5 is clocked by FOSC/4, Timer1/3/5 will not increment during Sleep. When the device wakes from Sleep, Timer1/3/5 will continue from its previous state.

Capture mode will operate during Sleep when Timer1/3/5 is clocked by an external clock source.

REGISTER 29-3: CCPRxL REGISTER: CCPx REGISTER LOW BYTE

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
CCPRxL<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Reset
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0

CCPxMODE = Capture mode
CCPRxL<7:0>: Captured value of TMR1/3/5L
CCPxMODE = Compare mode
CCPRxL<7:0>: LS Byte compared to TMR1/3/5L
CCPxMODE = PWM modes when CCPxFMT = 0
CCPRxL<7:0>: CCPW<7:0> – Pulse-width Least Significant eight bits
CCPxMODE = PWM modes when CCPxFMT = 1
CCPRxL<7:6>: CCPW<1:0> – Pulse-width Least Significant two bits
CCPRxL<5:0>: Not used.

REGISTER 29-4: CCPRxH REGISTER: CCPx REGISTER HIGH BYTE

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
CCPRxH<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Reset
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0

CCPxMODE = Capture mode
CCPRxH<7:0>: Captured value of TMR1/3/5H
CCPxMODE = Compare mode
CCPRxH<7:0>: MS Byte compared to TMR1/3/5H
CCPxMODE = PWM modes when CCPxFMT = 0
CCPRxH<7:2>: Not used
CCPRxH<1:0>: CCPW<9:8> – Pulse-width Most Significant two bits
CCPxMODE = PWM modes when CCPxFMT = 1
CCPRxH<7:0>: CCPW<9:2> – Pulse-width Most Significant eight bits

30.5.4 SLAVE MODE 10-BIT ADDRESS RECEPTION

This section describes a standard sequence of events for the MSSPx module configured as an I^2C slave in 10-bit Addressing mode.

Figure 30-20 is used as a visual reference for this description.

This is a step-by-step process of what must be done by slave software to accomplish I^2C communication.

1. Bus starts Idle.
2. Master sends Start condition; S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Master sends matching high address with R/$\overline{W}$ bit clear; UA bit of the SSPxSTAT register is set.
4. Slave sends $\overline{ACK}$ and SSPxIF is set.
5. Software clears the SSPxIF bit.
6. Software reads received address from SSPxBUF clearing the BF flag.
7. Slave loads low address into SSPxADD, releasing SCL.
8. Master sends matching low address byte to the slave; UA bit is set.

Note: Updates to the SSPxADD register are not allowed until after the $\overline{ACK}$ sequence.

9. Slave sends $\overline{ACK}$ and SSPxIF is set.

Note: If the low address does not match, SSPxIF and UA are still set so that the slave software can set SSPxADD back to the high address. BF is not set because there is no match. CKP is unaffected.

10. Slave clears SSPxIF.
11. Slave reads the received matching address from SSPxBUF clearing BF.
12. Slave loads high address into SSPxADD.
13. Master clocks a data byte to the slave and clocks out the slaves $\overline{ACK}$ on the ninth SCL pulse; SSPxIF is set.
14. If SEN bit of SSPxCON2 is set, CKP is cleared by hardware and the clock is stretched.
15. Slave clears SSPxIF.
16. Slave reads the received byte from SSPxBUF clearing BF.
17. If SEN is set the slave sets CKP to release the SCL.
18. Steps 13-17 repeat for each received byte.
19. Master sends Stop to end the transmission.

30.5.5 10-BIT ADDRESSING WITH ADDRESS OR DATA HOLD

Reception using 10-bit addressing with AHEN or DHEN set is the same as with 7-bit modes. The only difference is the need to update the SSPxADD register using the UA bit. All functionality, specifically when the CKP bit is cleared and SCL line is held low are the same. Figure 30-21 can be used as a reference of a slave in 10-bit addressing with AHEN set.

Figure 30-22 shows a standard waveform for a slave transmitter in 10-bit Addressing mode.

REGISTER 30-7: SSPxBUF: MSSP BUFFER REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
SSPxBUF<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **SSPxBUF<7:0>:** MSSP Buffer bits

TABLE 30-3: SUMMARY OF REGISTERS ASSOCIATED WITH MSSPx

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	—(3)	TRISA2	TRISA1	TRISA0	143
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	144
INLVLA(1)	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	146
TRISB(2)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	149
ANSELB(2)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	150
INLVLB(2)	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	152
TRISC	TRISC7(2)	TRISC6(2)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	155
ANSELC	ANSC7(2)	ANSC6(2)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	157
INLVLC(1)	INLVLC7(2)	INLVLC6(2)	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	159
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	100
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	107
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	102
PIR2	TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BCL2IF	TMR4IF	NCO1IF	108
PIE2	TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BCL2IE	TMR4IE	NCO1IE	103
SSPxSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	359
SSPxCON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				360
SSPxCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	362
SSPxCON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	363
SSPxMSK	SSPxMSK<7:0>								364
SSPxADD	SSPxADD<7:0>								364
SSPxBUF	SSPxBUF<7:0>								365
SSPxCLKPPS	—	—	—	SSPxCLKPPS<4:0>					162
SSPxDATPPS	—	—	—	SSPxDATPPS<4:0>					162
SSPxSSPPS	—	—	—	SSPxSSPPS<4:0>					162

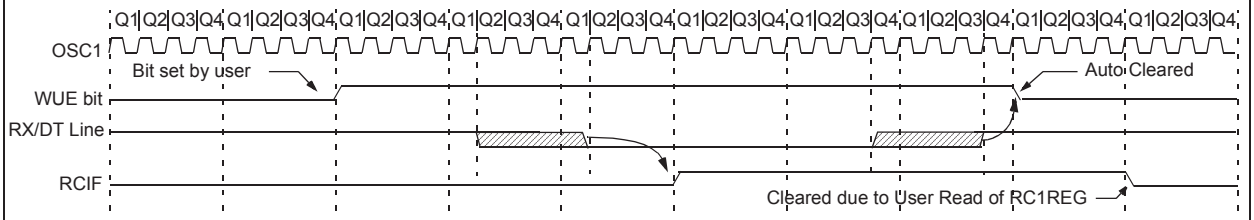
Legend: — = Unimplemented location, read as '0'. Shaded cells are not used by the MSSP module

Note 1: When using designated I^2C pins, the associated pin values in INLVLx will be ignored.

2: PIC16(L)F18346 only.

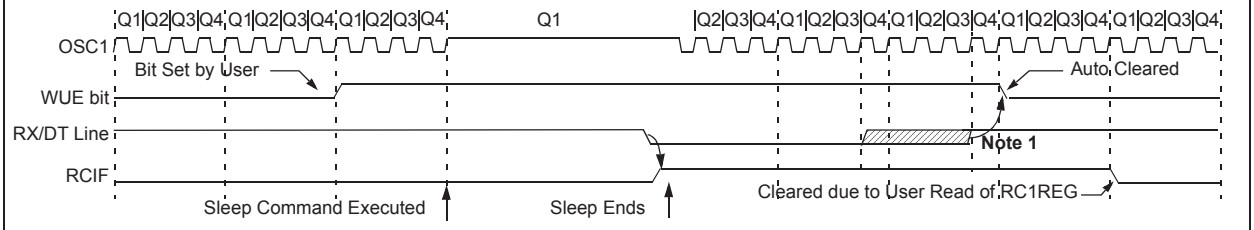
3: Unimplemented, read as '1'.

FIGURE 31-7: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION



Note 1: The **EUSART1** remains in Idle while the WUE bit is set.

FIGURE 31-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP



Note 1: If the wake-up event requires long oscillator warm-up time, the automatic clearing of the WUE bit can occur while the `stposc` signal is still active. This sequence should not depend on the presence of Q clocks.

2: The **EUSART1** remains in Idle while the WUE bit is set.

31.3.4 BREAK CHARACTER SEQUENCE

The EUSART1 module has the capability of sending the special Break character sequences that are required by the LIN bus standard. A Break character consists of a Start bit, followed by 12 '0' bits and a Stop bit.

To send a Break character, set the SENDB and TXEN bits of the TX1STA register. The Break character transmission is then initiated by a write to the TX1REG. The value of data written to TX1REG will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

The TRMT bit of the TX1STA register indicates when the transmit operation is active or idle, just as it does during normal transmission. See Figure 31-9 for the timing of the Break character sequence.

31.3.4.1 Break and Sync Transmit Sequence

The following sequence will start a message frame header made up of a Break, followed by an auto-baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART1 for the desired mode.
2. Set the TXEN and SENDB bits to enable the Break sequence.
3. Load the TX1REG with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TX1REG to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware and the Sync character is then transmitted.

When the TX1REG becomes empty, as indicated by the TXIF, the next data byte can be written to TX1REG.

31.3.5 RECEIVING A BREAK CHARACTER

The Enhanced EUSART1 module can receive a Break character in two ways.

The first method to detect a Break character uses the FERR bit of the RC1STA register and the received data as indicated by RC1REG. The Baud Rate Generator is assumed to have been initialized to the expected baud rate.

A Break character has been received when:

- RCIF bit is set
- FERR bit is set
- RC1REG = 00h

The second method uses the Auto-Wake-up feature described in **Section 31.3.3 "Auto-Wake-up on Break"**. By enabling this feature, the EUSART1 will sample the next two transitions on RX/DT, cause an RCIF interrupt, and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Detect feature. For both methods, the user can set the ABDEN bit of the BAUD1CON register before placing the EUSART1 in Sleep mode.

FIGURE 31-9: SEND BREAK CHARACTER SEQUENCE